

application program that can read that data. The unknown data can be maintained by the application program as discussed above with reference to **FIG. 5**. The method then proceeds to step **630** where items in the modified original data file are converted to the format of the previous version(s) and saved. Thus, the modified original data file is saved in the formats of both the active version and the previous version(s).

[**0071**] Next, the method determines whether the low version watermark needs to be updated. In step **650**, the active version of the application program is compared to the low version watermark of the original data file. If it is determined in step **650** that the active version is older than the version of the application program represented by the low version watermark, then the method branches to step **655** where the low version watermark is updated to correspond to the active version. The method then proceeds to step **665** where the last version to save the file is also updated for future use. If it is determined in step **650** that the active version is the same as or newer than the version of the application program represented by the low version watermark, then the method branches directly to step **665** where the last version watermark is updated to correspond to the active version.

[**0072**] If the comparison of step **620** indicates that the high version watermark represents a version of the application program that is the same as the active version, then the original data file corresponds to the active version of the application program. In that case, the method branches to step **660** where the modified original data file is saved by overwriting the original data file. The method then proceeds to step **630** where items in the modified original data file are converted to the format of the previous version(s) and saved. Thus, the modified original data file is saved in the formats of both the active version and the previous version(s). The method then proceeds to step **665** where the last version to save the file is updated for future use.

[**0073**] Note that step **630** can be performed regardless of the branch taken from step **620**. Accordingly, properties corresponding to previous versions can always be written to the file so that that file is compatible with previous versions.

[**0074**] In some cases, methods **500** and **600** can comprise an additional step (not shown) of tracking the location of OPL array elements to save the file after it has been modified. For example, a version of a data file may comprise a table having four elements. However, the modified version of the data file may have only three elements. When the application program attempts to overwrite the original data file, the disk version includes the original four elements, but the memory version includes only three elements. Without the tracking step, the application program may not know which element has been deleted. In the tracking step, the application program tracks the original index of all tables. When a table is read from disk, an extra data item indicating the position of each element in the table is added. In other words, the application program marks which element was first, second, etc. Accordingly, the program determines which element has been deleted because it does not exist and can preserve future version properties that might exist in the array elements.

[**0075**] In methods **500** and **600**, the active version of the application program can use any of the watermarks to

determine what actions to take when loading the original data file. Use of the high version watermark has been discussed above. The low version watermark can give a hint to the application program and limits the types of conversion necessary on the original data file. Building on a previous example, if the hyperlink table in version 1.0 is known not to exist, because the low version watermark indicates the file has never been written by that version, the conversion process can safely skip anything that deals with the hyperlink table. The last version watermark can indicate data that is known by the last version to write the file, as well as data that is unknown but carried in the file.

[**0076**] The object version watermark can provide specific information for a particular object, whereby the particular object can be loaded, saved, or converted based on that information. The object version watermark can provide information for the particular object and can indicate the highest file version of any property that a specific object contains. As an example, suppose that a hyperlink property was added to object X in version 2.0, and no new properties were added to object X in version 3.0. If object X contains a hyperlink written in version 2.0 or a later version, its version watermark would be set to version 2.0. If object X does not contain a hyperlink (and therefore no hyperlink property is written) or was written by version 1.0, the version watermark would be set to version 1.0. This information can be used to further optimize loading and saving operations with respect to each individual object and the highest version that would understand all properties in that object. For example, object X that was written by version 3.0 and did not contain a hyperlink (and would therefore have a version 1.0 object watermark) could be written and read by version 1.0 without being concerned about future version properties existing, even though the file high version watermark would reflect version 3.0.

[**0077**] Generally, the most current version of the application program to save a file is best suited to determine how to handle any information from that version or previous versions. The most current version can read and understand everything in the file. The most current version of the application program determines which information to keep, convert, or discard. Accordingly, the highest version to write a file determines whether information is discarded or converted. All lower versions can simply propagate any unknown information.

[**0078**] Finally, all steps of methods **500** and **600** shown in **FIGS. 5 and 6** are not required for the invention to be operable. Additionally, some steps may be performed in a different order than illustrated.

[**0079**] With reference to **FIGS. 7-9**, specific exemplary embodiments of methods **500** and **600** according to the present invention will be described. In each of **FIGS. 7-9**, method **500** or **600** has determined that the last version watermark is less than the active version of the application program (i.e., that the last version of the application program to save the original data file corresponds to a previous version of the application program with respect to the active version) (see step **595** of **FIG. 5**). Additionally, the embodiments described below with reference to **FIGS. 7-9** are also operable when method **500** or **600** has determined that the high version watermark is less than the active version of the application program.